Assignment 1

Derek Paulsen

1 Problem 1

1.1 Part A

In order to describe a single hash function we need to store three numbers a, b and p, where, $p > |U|, 1 \geq a < p, 0 \geq b < p$, therefore we need to at most $\log_2(|U|)$ bits to store each number. Hence a single hash function requires $O(\log_2(|U|))$ bits. We need $n + 1$ hash functions to describe the two level hashing hence we need $O(n \log_2(|U|))$ to describe the perfect two level hashing function.

1.2 Part B

Let the items that we are trying to hash be $S = (s_1, ..., s_n)$. (the items of S are inserted in order). Suppose we have a single array of size $O(n^{1.5})$. Let $X_i = \mathbb{1}[s_i$ collides with another element in the array], we wish to compute $E[X_i]$. Let h be a random hash function $h(x) \to \{1, ..., n^{1.5}\}$

$$E[X_i] = E[\sum_{j=1}^{i-1} \mathbb{1}[h(s_i) = h(s_j)]]$$

$$E[X_i] = \sum_{j=1}^{i-1} E[\mathbb{1}[h(s_i) = h(s_j)]]$$

$$E[X_i] = \sum_{j=1}^{i-1} Pr[h(s_i) = h(s_j)]$$

$$E[X_i] = \sum_{j=1}^{i-1} 1/n^{1.5}$$

$$E[X_i] = (i-1)/n^{1.5}$$

suppose we have a two arrays of size $O(n^{1.5})$. The probability of a collision in this both arrays is, Let $Y_i = \mathbb{1}[s_i$ collides with another element in both arrays], we wish to compute $E[Y_i]$. We note that each array has at most $i - 1$ elements when inserting the ith element, hence we have

$$E[Y_i] \leq E[X_i]^2$$

$$E[Y_i] \leq (i-1)^2/n^3$$

Let $Y = \sum_{i=1}^{n} = Y_i$, the number of collisions. We want $E[Y]$

$$
\begin{aligned}
E[Y_i] &= E[\sum_{i=1}^{n} Y_i] \\
E[Y_i] &= \sum_{i=1}^{n} E[Y_i] \\
E[Y_i] &\leq \sum_{i=1}^{n} \frac{(i-1)^2}{n^3} \\
E[Y_i] &\leq \frac{n(n-1)(2n-1)}{6n^3} \\
E[Y_i] &\leq \frac{n^3 - 3n^2 + n}{6n^3} \\
E[Y_i] &= O(1)
\end{aligned}
$$

Therefore the expected number of collisions is $O(1)$

1.3 Part C

We note that our work above only assumes that $x, y \in U, x \neq y, Pr[h(x) = h(y)] = 1/n^{1.5}$, which is satisfied by a 2-universal hash function by definition. Therefore the expected number of collisions is still $O(1)$ when the hash functions 2-universal hash functions.

1.4 Part D

The algorithm procedes as follows,

1. Hash all elements $s \in S$ into the first array with collisions
2. remove elements from the first array such that each bucket has at most one element, let C be the set of elements that were removed
3. hash the elements C into the second array
4. repeat step 3 until a perfect hashing has been found for the elements in C

Let $X = \sum_{i=1}^{n} X_i$, we wish to compute $E[X]$ the number of expected collisions.

$$
\begin{aligned}
E[X] &= E[\sum_{i=1}^{n} X_i] \\
E[X] &= \sum_{i=1}^{n} E[X_i] \\
E[X] &= \sum_{i=1}^{n} (i-1)/n^{1.5} \\
E[X] &= \sum_{i=1}^{n} \frac{(n-1)(n-2)}{n^{1.5}} \\
E[X] &= \frac{n^2 - 3n + 2}{n^{1.5}} \\
E[X] &\leq \sqrt{n}
\end{aligned}
$$

We now compute the probability of collision in step 3, using the union bound we get

$$
\begin{aligned}
Pr[collision] &\leq \sum_{i=1}^{\sqrt{n}} X_i \\
Pr[collision] &\leq \frac{(\sqrt{n}-1)(\sqrt{n}-2)}{n^{1.5}} \\
Pr[collision] &\leq \frac{n-3\sqrt{n}+2}{n^{1.5}} \\
Pr[collision] &\leq 1/\sqrt{n}
\end{aligned}
$$

Therefore in expectation we will need try one hash function in step 3 to get a perfect hashing for C.
Step 1 requires n hash functions evaluations, and step 3 requires $\sqrt{n}$ hash functions evaluations in expectation, hence the number of hash function evaluations to get a perfect hashing with the above algorithm is $O(n)$.

1.5 Part E

2 Problem 2

2.1 Part A

There are n choose k ways to have k balls in the first bucket and then $(n-1)^{n-k}$ ways to place the rest of the balls. There are n^n total ways to place all the balls, hence we want,

$$
\begin{aligned}
\Pr[\text{bin 1 has } k \text{ balls}] &\geq 1/\sqrt{n} \\
\frac{\binom{n}{k}(n-1)^{n-k}}{n^n} &\geq 1/\sqrt{n}
\end{aligned}
$$

Using the fact that $(\frac{n}{k})^k \leq \binom{n}{k}$

$$
\begin{aligned}
\frac{(\frac{n}{k})^k(n-1)^{n-k}}{n^n} &\geq 1/\sqrt{n} \\
\frac{1}{k^k} &\geq \frac{(\frac{n}{n-1})^{n-k}}{\sqrt{n}}
\end{aligned}
$$

using this bound we get $(1+\frac{1}{n-1})^n \leq e \implies (1+\frac{1}{n-1})^{n-k} \leq e$

$$
\begin{aligned}
\frac{1}{k^k} &\geq \frac{e}{\sqrt{n}} \\
k^k &\leq \frac{\sqrt{n}}{e}
\end{aligned}
$$

We now set $k = \log(n)/e\log\log(n)$,

$$
\begin{aligned}
(\log n/e\log\log n)^k &\leq \frac{\sqrt{n}}{e} \\
k(\log\log n - \log\log\log n - 1) &\leq .5\log n - 1 \\
\frac{1}{e}\log n - k(\log\log\log n + 1) &\leq .5\log n - 1 \\
&True
\end{aligned}
$$

Therefore we have that the probability of bin 1 having k balls is greater than or equal to $1/\sqrt{n}$

2.2 Part B

I don't have a formal proof for this, however here is the idea, Let a_i be the event that bin i has k balls. Let b_i be the event that at least one bin $i-1,...,1$ has k balls. We then have the probability that bin i has k given that bin $i-1,...,1$ dont have k is,

$$Pr(a_i|\bar{b}_{i-1}) = \frac{Pr(\bar{b}_{i-1}|a_i)Pr(a_i)}{1 - Pr(b_{i-1})}$$

We then note that $Pr(b_i)$ is strictly increasing while $Pr(a_i)$ is constant, therefore, $Pr(a_i|\bar{b}_{i-1})$ is strictly increasing.
We then have the probability that none of the bins have k is,

$$Pr(\bigcap_{i=1}^{n} \bar{a}_i) = \prod_{i=1}^{n} Pr(\bar{a}_i| \bigcap_{j=1}^{i-1} \bar{a}_j)$$
$$Pr(\bigcap_{i=1}^{n} \bar{a}_i) = \prod_{i=1}^{n} 1 - Pr(a_i| \bigcap_{j=1}^{i-1} \bar{a}_j)$$
$$Pr(\bigcap_{i=1}^{n} \bar{a}_i) = \prod_{i=1}^{n} 1 - Pr(a_i|\bar{b}_{i-1})$$
$$Pr(\bigcap_{i=1}^{n} \bar{a}_i) \leq \prod_{i=1}^{n} 1 - 1/\sqrt{n}$$
$$Pr(\bigcap_{i=1}^{n} \bar{a}_i) \leq (1 - 1/\sqrt{n})^n$$
$$Pr(\bigcap_{i=1}^{n} \bar{a}_i) \leq (1 - 1/\sqrt{n})^n$$
$$Pr(\bigcap_{i=1}^{n} \bar{a}_i) \leq ((1 - 1/\sqrt{n})^{\sqrt{n}})^{\sqrt{n}}$$
$$Pr(\bigcap_{i=1}^{n} \bar{a}_i) \leq 1/e^{\sqrt{n}}$$

We then use that fact that $e^{\sqrt{n}} \geq n$

$$Pr(\bigcap_{i=1}^{n} \bar{a}_i) \leq 1/n$$

Therefore we have that the probability of some bin having k is greater than or equal to $1 - 1/n$.

3 Problem 3

Suppose that we have the a sample of t numbers from the unordered list. Let $p = k/n$, i.e. the percentile of the rank that we are trying to estimate. Our estimator is the number with rank tp in our sample.
Proof.
We begin by noting that our esimator is incorrect if and only if we have tp elements which are less than $(1-\epsilon)k$ or we have $t(1-p)$ elements which are greater $(1+\epsilon)k$. Let $S = \{s_i, ..., s_t\}$ be a random sample with replacement of size t from $x_1, ...x_n$. Let $L_i = \mathbb{1}[rank(s_i) < (1-\epsilon)k]$ and $L = \sum_{i=1}^{t} L_i$. We then compute the expectation,

$$
\begin{aligned}
E[L] &= E[\sum_{i=1}^{t} L_i] \\
&= \sum_{i=1}^{t} E[L_i] \\
&= \sum_{i=1}^{t} (1-\epsilon)p \\
&= t(1-\epsilon)p
\end{aligned}
$$

Additionally, let $U_i = \mathbb{1}[rank(s_i) > (1+\epsilon)k]$ and $U = \sum_{i=1}^{t} U_i$. We then compute the expectation,

$$
\begin{aligned}
E[U] &= E[\sum_{i=1}^{t} U_i] \\
&= \sum_{i=1}^{t} E[U_i] \\
&= \sum_{i=1}^{t} (1-(1+\epsilon)p) \\
&= t(1-(1+\epsilon)p)
\end{aligned}
$$

We then have

$$
Pr[(1-\epsilon)k > rank(k) \vee rank(k) < (1=\epsilon)k] \leq Pr[L \geq tp] + Pr[U \geq t(1-p)]
$$

We now compute bounds on the probabilities. We note that $(1+\frac{\epsilon}{1-\epsilon})t(1-\epsilon)p = tp$

$$
Pr[L \geq (1+\frac{\epsilon}{1-\epsilon})t(1-\epsilon)p] \leq e^{\frac{-(\frac{\epsilon}{1-\epsilon})^2 t(1-\epsilon)p}{2+\frac{\epsilon}{1-\epsilon}}}
$$

We then set this to δ

$$
\begin{aligned}
\delta &= e^{\frac{-(\frac{\epsilon}{1-\epsilon})^2 t(1-\epsilon)p}{2+\frac{\epsilon}{1-\epsilon}}} \\
log(1/\delta) &= t\frac{(\frac{\epsilon}{1-\epsilon})^2(1-\epsilon)p}{2+\frac{\epsilon}{1-\epsilon}} \\
\frac{log(1/\delta)(2+\frac{\epsilon}{1-\epsilon})}{(\frac{\epsilon}{1-\epsilon})^2(1-\epsilon)p} &= t \\
\frac{log(1/\delta)(2+\frac{\epsilon}{1-\epsilon})(1-\epsilon)}{\epsilon^2 p} &= t \\
\frac{log(1/\delta)(2(1-\epsilon)+\epsilon)}{\epsilon^2 p} &= t \\
\frac{log(1/\delta)(2-\epsilon)}{\epsilon^2 p} &= t
\end{aligned}
$$

we then do the same for the other element, We now compute bounds on the probabilities. We note that $(1+\frac{\epsilon p}{1-(1+\epsilon)p})t(1-(1+\epsilon)p) = t(1-p)$

$$Pr[U \geq (1 + \frac{\epsilon p}{1-(1+\epsilon)p})t(1-(1+\epsilon)p)] \leq e^{\frac{-(\frac{\epsilon p}{1-(1+\epsilon)p})^2 t(1-(1+\epsilon)p)}{2+\frac{\epsilon p}{1-(1+\epsilon)p}}}$$

We then set this to δ

$$\begin{aligned}
\delta &= e^{\frac{-(\frac{\epsilon p}{1-(1+\epsilon)p})^2 t(1-(1+\epsilon)p)}{2+(\frac{\epsilon p}{1-(1+\epsilon)p})}} \\
log(1/\delta) &= \frac{(\frac{\epsilon p}{1-(1+\epsilon)p})^2 t(1-(1+\epsilon)p)}{2+(\frac{\epsilon p}{1-(1+\epsilon)p})} \\
\frac{log(1/\delta)(2+\frac{\epsilon p}{1-(1+\epsilon)p})}{(\frac{\epsilon p}{1-(1+\epsilon)p})^2(1-(1+\epsilon)p)} &= t \\
\frac{log(1/\delta)(2+\frac{\epsilon p}{1-(1+\epsilon)p})(1-(1+\epsilon)p)}{\epsilon^2 p^2} &= t \\
\frac{log(1/\delta)(2-2p-\epsilon p)}{\epsilon^2 p^2} &= t
\end{aligned}$$

Now combine the two probabilities,

$$\begin{aligned}
t &= \frac{log(1/\delta)(2-2p-\epsilon p)}{\epsilon^2 p^2} + \frac{log(1/\delta)(2-\epsilon)}{\epsilon^2 p} \\
t &= log(1/\delta)(\frac{(2-2p-\epsilon p)}{\epsilon^2 p^2} + \frac{(2-\epsilon)}{\epsilon^2 p}) \\
t &= \frac{log(1/\delta)2(1-\epsilon p)}{\epsilon^2 p^2} \\
t &= \frac{log(1/\delta)2(1-\epsilon\frac{k}{n})n^2}{\epsilon^2 k^2} \\
t &= \frac{log(1/\delta)2(n^2-\epsilon nk)}{\epsilon^2 k^2}
\end{aligned}$$

Therefore we need $O(\frac{log(1/\delta)2(n^2-\epsilon nk)}{\epsilon^2 k^2})$ queries to distingush

4 Problem 4

4.1 Part A

Suppose that we we randomly create a set $Q \subseteq \{1, ..., n\}$ where each element has $1/k$ chance of being included in the set. We first consider $Pr[Q \cap S = \emptyset]$. Because of how we constructed the set, for each element $s \in S$, $Pr[s \notin Q] = 1 - 1/k$, therefore $Pr[Q \cap S = \emptyset] = (1-1/k)^{|S|}$. Let

$$X_i = \mathbb{1}[Q \cap S = \emptyset] \text{ and } X = \sum_{i=1}^{t} X_i \text{ and } p = (1-1/k) = \frac{k-1}{k}$$

We note that,

$$
\begin{aligned}
E[X] &= E[\sum_{i=1}^{t} X_i] \\
&= \sum_{i=1}^{t} E[X_i] \\
&= \sum_{i=1}^{t} (1-1/k)^{|S|} \\
E[X] &= t(1-1/k)^{|S|} \\
E[X] &= tp^{|S|}
\end{aligned}
$$

Additionally,

$$
\begin{aligned}
|S| \leq k &\implies E[X] \geq tp^k \\
|S| \geq (1+\epsilon)k &\implies E[X] \leq tp^{(1+\epsilon)k}
\end{aligned}
$$

When $X \leq (1-\epsilon/2)E[X]$ we predict that $|S| \geq (1+\epsilon)k$. Suppose that $|S| \leq k$ we know that the $\mu = E[X] \geq tp^k$. Therefore, using the Chernoff bound,

$$
Pr[X \leq (1-\epsilon)\mu] \leq e^{-\frac{\epsilon^2\mu}{4(2+\epsilon})}
$$

now if we set our probability of failure to be δ,

$$
\begin{aligned}
\delta &= e^{-\frac{\epsilon^2\mu}{4(2+\epsilon)}} \\
log(\delta) &= -\frac{\epsilon^2\mu}{2+\epsilon} \\
log(1/\delta) &= \frac{\epsilon^2\mu}{4(2+\epsilon)} \\
\frac{log(1/\delta)4(2+\epsilon)}{\epsilon^2} &= \mu \\
\frac{log(1/\delta)4(2+\epsilon)}{\epsilon^2} &= tp^k
\end{aligned}
$$

note that for $1 \geq k, p^k = (\frac{k-1}{k})^k \leq 1$

$$
\begin{aligned}
\frac{log(1/\delta)4(2+\epsilon)}{\epsilon^2} &= t \\
O(\frac{log(1/\delta)}{\epsilon^2}) &= t
\end{aligned}
$$

Now assume that $|S| \geq (1+\epsilon)k$, First note that

$$
(1+((1-\epsilon/2)p^{-k\epsilon}-1))p^{(1+\epsilon)k} = (1-\epsilon/2)p^k
$$

The probability of making an error is then,

$$
Pr[X \geq (1+((1-\epsilon/2)p^{-k\epsilon}-1))\mu] \leq e^{\frac{((1-\epsilon/2)p^{-k\epsilon}-1))^2\mu}{2+((1-\epsilon/2)p^{-k\epsilon}-1))}}
$$

We set $\mu = tp^{(1+\epsilon)k}$ since the closer the true μ is to the assumed $\mu = tp^k$ the more likely we are to make an error, setting this to δ we get

$$
\begin{aligned}
\delta &= e^{\frac{-((1-\epsilon/2)p^{-k\epsilon}-1))^2\mu}{2+((1-\epsilon/2)p^{-k\epsilon}-1))}} \\
log(1/\delta) &= \frac{((1-\epsilon/2)p^{-k\epsilon}-1))^2\mu}{2+((1-\epsilon/2)p^{-k\epsilon}-1))} \\
log(1/\delta)(2+((1-\epsilon/2)p^{-k\epsilon}-1))) &= ((1-\epsilon/2)p^{-k\epsilon}-1))^2\mu \\
\frac{log(1/\delta)(2+((1-\epsilon/2)p^{-k\epsilon}-1)))}{((1-\epsilon/2)p^{-k\epsilon}-1))^2} &= \mu \\
\frac{log(1/\delta)(2+((1-\epsilon/2)p^{-k\epsilon}-1)))}{((1-\epsilon/2)p^{-k\epsilon}-1))^2} &= tp^{(1+\epsilon)k} \\
\frac{log(1/\delta)(2+((1-\epsilon/2)p^{-k\epsilon}-1)))}{((1-\epsilon/2)p^{-k\epsilon}-1))^2p^{(1+\epsilon)k}} &= t \\
O(\frac{log(1/\delta)}{\epsilon^2}) &= t
\end{aligned}
$$

Therefore, in order to make the probability of error $1-\delta$ we only need $t = O(\frac{log(1/\delta)}{\epsilon^2})$ queries.

4.2 Part B

Let $F(k, k')$ be the estimator from part A, where the $F(k, k') = 0$ if the size of S is predicted to be less than or equal to k and $F(k, k') = 1$ if the size of S is predicted to be greater than or equal to k' (eqivalent to $(1+\epsilon)k$). Our algorithm is as follows,

1. $l \leftarrow 1$
2. $u \leftarrow n$
3. while $l > (1+\epsilon)u$
   (a) // The quartiles for the current range
   (b) $q_1 \leftarrow l + (u-l)/4$, $q_2 \leftarrow l + (u-l)/2$, $q_3 \leftarrow l + 3(u-l)/4$
   (c) If $F(q_1, q_2) \neq F(q_2, q_3)$ then $l \leftarrow q_1, u \leftarrow q_3$ // Take the second and third quartile
   (d) If $F(q_1, q_2) = F(q_2, q_3) = 1$ then $l \leftarrow q_2, u \leftarrow u$ // Take the bottom half
   (e) If $F(q_1, q_2) = F(q_2, q_3) = 0$ then $l \leftarrow l, u \leftarrow q_2$ // Take the top half
4. Return l

4.2.1 Probability

Assuming $l \leq |S| \leq u$ we have 4 cases, one for each quartile. If $|S|$ is in the first quartile, then with probability $(1-\delta)^2$ we have $F(q_1, q_2) = F(q_2, q_3) = 0$, in which case $|S|$ stays in the range. Similar reasoning can be applied if $|S|$ is in the fourth quartile. If $|S|$ is in the second quartile, then $F(q_2, q_3)$ is reliable but $F(q_1, q_2)$ is not. However, as long as $F(q_2, q_3)$ is correct, $|S|$ stays in the range. Symmetric reasoning is applied for when $|S|$ is in the third quartile. Therefore the probability of success for a given iteration is at least $(1-\delta)^2$.
In order to succeed we must have every iteration succeed, hence the probability that the algorithm succeeds is at least $(1-\delta)^{2\log_2 n}$.

4.2.2 Runtime

We note that the size of the interval we consider halves, therefore, for any ϵ we perform at most $O(\log n)$ iterations. Each iteration takes a constant number of estimator evaluations therefore the number of $O(\log n)$ number of queries. We want our probability of success to be at least $2/3$ hence we get,

$$
\begin{aligned}
(1-\delta)^{2\log n} &\geq 2/3 \\
\log(1-\delta)(2\log n) &\geq \log(2/3) \\
\log(1-\delta) &\geq \frac{\log(2/3)}{(2\log n)} \\
1-\delta &\geq (2/3)^{1/2\log n} \\
\delta &\leq 1-(2/3)^{1/2\log n} \\
\log(1/\delta) &\geq -\log(1-(2/3)^{1/2\log n}) \\
\log(1/\delta) &= O(\log\log x)
\end{aligned}
$$

Since the number of queries per estimator evaluation is $O(\log(1/\delta)/\epsilon^2)$ the number of queries per iteration is $O(\log\log x)$. Therefore our runtime complexity is $O(\log n \log\log n)$.

4.3 Part C

We can randomly sample a unique element from S as follows,

- $Q = \{1, ..., n\}$
- While $|Q| > 1$
  - Q' = random subset of Q containing each element with probability $1/2$
  - While $Q' \cap S = \emptyset$
    - Q' = random subset of Q containing each element with probability $1/2$
  - $Q = Q'$
- Return the single element in Q

4.3.1 Runtime

First we note that Q always contains at least one element in S. Because the way Q' is constructed, $Pr[Q' \cap S = \emptyset] \leq 1/2$ at every step of the algorithm. Therefore we have that each iteration in the worst case (where S only has one element) we expect two queries to be executed. We then note that the size of Q halves in expectation each iteration, therefore there are $O(\log_2 n)$ expected iterations required. Since the expected number of queries per iteration is constant, the runtime of the algorithm is $O(\log_2 n)$

4.3.2 Sampling Randomness

Suppose our algorithm runs for h steps before $|Q \cap S| = 1$ at which point the output of the algorithm is determined and with the single element. We note that each element in each iteration has $1/2$ probability of being include in the next iteration, therefore the probability of each element being in Q after h iterations is $1/2^h$. Since all elements have equal probability of being in the final set, the sample is uniformly random.